



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Tony Puharinen

ETÄOHJELMOINTIYMPÄRISTÖN LUOMINEN

Tekniikka
2017

TIIVISTELMÄ

Tekijä	Tony Puharinen
Opinnäytetyön nimi	Etäohjelmointiympäristön luominen
Vuosi	2017
Kieli	suomi
Sivumäärä	(40 + 1 liite)
Ohjaaja	Mika Billing

Tämän opinnäytetyön tavoitteena oli kehittää etäohjelmointiympäristö, joka jäljittelee Technobothniassa sijaitsevaa robottisolua. Etäohjelmointiympäristöjä käyttämällä tarpeettomat liikkeet, läheltä piti -tilanteet ja nivelratojen ylitykset huomataan ajoissa ja täten saavutetaan tarkempia ohjelmia.

Työhön kuului robottisolun komponenttien mitoittaminen ja mallintaminen, layoutin rakentaminen ja ympäristön saattaminen ohjelmointivalmiuteen. Ympäristön on tarkoitus tulla käyttöön opetuksen yhteydessä.

Kehityksen lähtökohtana oli saada aikaan tarkka ja realistisen näköinen ympäristö, joka on muutettavissa tarpeen mukaan. Valmiissa solussa on kaikki samat komponentit, jotka löytyivät oikeasta solustakin työn valmistuttua. Soluun on sittemmin tullut muutoksia.

ABSTRACT

Author	Tony Puharinen
Title	Creating an Offline Programming Environment
Year	2017
Language	Finnish
Pages	(40 + 1 Appendix)
Name of Supervisor	Mika Billing

The objective of this thesis was to develop an offline programming environment that emulates a robot cell located in the Technobothnia research laboratories. With the use of offline programming environment, unnecessary movements, narrow escapes and exceeded joint limits are noticed early on and thus more accurate programs are achieved.

The thesis involved measuring and modeling the robotic cell components, building the layout and bringing the environment operational. The environment is intended to be introduced to educational training.

The basis of the development was to generate an accurate and realistic looking environment that is changeable when needed. The completed cell has all the same components, as the real cell it is based upon at the time of completion. The cell has undergone changes since.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO	9
1.1	Työn tavoitteet	9
1.2	Yrityskatsaus	9
1.2.1	VAMK	10
1.2.2	Technobothnia	10
2	TAUSTATIETOA OPINNÄYTETYÖN AIHEESTA	11
2.1	Teollisuusrobotti	11
2.1.1	Robottityypit	12
2.1.2	Robottien työkalut	12
2.1.3	Ohjausjärjestelmä	12
2.1.4	Koordinaatistot	13
2.1.5	Ohjelmointi	14
2.2	3D-CAD	15
2.2.1	3D-ohjelmistot	15
2.2.2	Mallintaminen	17
2.2.3	3D-mallit	18
2.3	Simulaatio ja virtuaalitodellisuus	18
2.3.1	Simulaatio ja virtuaalitodellisuus robotiikassa	19
2.3.2	Simulointiympäristö	20
2.3.3	Virtuaaliympäristö	21
3	SOLUN ETÄOHJELMOINTIYMPÄRISTÖ	22
3.1	Robottisolun mallintaminen	22
3.1.1	Mitoittaminen	22
3.1.2	3D-CAD -mallit	22
3.1.3	Ratkaisut mallien keventämiseksi	25
3.2	CAD -mallien siirto etäohjelmointiympäristöön	26
3.2.1	Komponenttien viimeistely	26

3.2.2	Layout	30
3.3	Ympäristön kalibrointi	36
3.4	Etäohjelmointiympäristön siirto.....	37
4	POHDINTAA.....	38
4.1	Työkatsaus	38
4.2	Jatkokehitys.....	39
	LÄHTEET	40

LIITTEET

KUVALUETTELO

Kuva 1. Opetus- ja tutkimuslaboratorio Technobothnia.	9
Kuva 2. Yleisimmät teollisuusrobotit.	11
Kuva 3. Robotiikassa käytetyt koordinaatistot.	13
Kuva 4. Esimerkki Rapid -ohjelmointikielestä. /6/	15
Kuva 5. Siemens NX 10.0 -mallinnusohjelmisto.	16
Kuva 6. Koordinaatistojen keskipiste, eli origo.	17
Kuva 7. HTC Vive -virtuaalitodellisuuspaketti.	19
Kuva 8. RobotStudion simulointiympäristö.	20
Kuva 9. RobotStudion virtuaaliympäristö.	21
Kuva 10. Siemens NX 10.0:n toimintovalikko.	23
Kuva 11. Mallinnetun solun aitaus.	23
Kuva 12. Yksi mallinnetun solun kuljettimista.	24
Kuva 13. Mallinnetun solun turvakytin.	25
Kuva 14. Työkalupisteen kalibrointiprosessi.	26
Kuva 15. Työkalutietojen lataaminen robotilta.	27
Kuva 16. Ote työkalutiedoista.	28
Kuva 17. Työkalutietojen syöttäminen toimilaitteen mallille.	29
Kuva 18. Työkalupisteiden koordinaatit toimilaitteen mallissa.	29
Kuva 19. Import library.	30
Kuva 20. Soluun lisätty robotin jalusta.	31
Kuva 21. Loput komponentit lisättynä soluun.	31
Kuva 22. ABB Library.	32
Kuva 23. Robotin kapasiteetin ja ulottuvuuden valinta.	32
Kuva 24. Lisätyn robotin paikoittaminen 1/2.	33
Kuva 25. Lisätyn robotin paikoittaminen 2/2.	34
Kuva 26. VirtualControllerin luominen.	34
Kuva 27. Controllerin toiminnan ilmoittava ikkuna.	35
Kuva 28. Maailmankoordinaatiston keskelle ilmestynyt työkalu.	35
Kuva 29. Työkalu kiinnitettynä robottiin.	36
Kuva 30. Käyttäjäkoordinaatisto.	37
Kuva 31. Rakennettu virtuaalinen solu ja oikea solu vierekkäin.	38

LIITELUETTELO

LIITE 1. Komponenttien paikoitukset solun maailmankoordinaatistossa.

KÄYTETYT LYHENTEET JA TERMIT

CAD	Tietokoneavusteinen suunnittelu.
DOF	<i>Degrees of Freedom</i> . Vapausaste.
TKP	Työkalupiste. Työkalun kärjessä olevan koordinaattiston origo.
STEP	Standardisoitu tuotetiedon tiedostomuoto. (ISO 10303)
Automaatio	Tietokoneen hyödyntäminen laitteiston itsetoimivuuden saavuttamiseksi.
Etäohjelmointiympäristö	Rakennettu ympäristö, jota käytetään robotin etäohjelmoinnin apuna.
Komponentti	Mallinnettu kokonaisuus, joka jäljittelee olemassa olevaa rakennetta.
Layout	Sommittelu tai pohjapiirustus.
Origo	Koordinaatiston keskellä oleva piste, jonka arvot ovat: 0, 0, 0.
Parametri	Numerolla tai lausekkeella ilmaistu arvo.
Solu	Suljettu robotin toiminta-alue.

1 JOHDANTO

1.1 Työn tavoitteet

Työn tavoite on kehittää robotiikan koulutusohjelmaa Vaasan korkeakoululle luomalla etäohjelmointiympäristö RobotStudiolla, joka jäljittelee olemassa olevaa robottisolua. RobotStudio -ohjelmistoa on käytetty aktiivisesti opetuksen yhteydessä vuodesta 2014 lähtien sen monipuolisten ominaisuuksien johdosta, mutta kunnollisen simulaatioympäristön puuttuessa opiskelijat ovat käyttäneet ns. luurankomalleja hankkeiden ja harjoitustöiden demoissa. Vaikka luurankomallien käyttö saattaa johtaa toimiviin ratkaisuihin, niin kehitetty etäohjelmointiympäristö tarjoaa edun tarkkuudessa ja erityisesti esitystilanteessa.

Kehittämällä etäohjelmointiympäristöä robotin simuloiminen helpottuu, koska esimerkiksi robotin ulottuvuus ja törmäysvaarat saadaan huomioitua ennen ohjelman siirtämistä fyysiselle robotille. Työllä on myös suora vaikutus opetukseen, koska uusi layout tarjoaa uusia opetuspisteitä simuloinnin ja virtuaalitodellisuuden parissa, joten työryhmien kokoa voidaan pienentää.

1.2 Yrityskatsaus

Opinnäytetyö suoritettiin kesällä 2017 Technobothnian robottilaboratoriossa.



Kuva 1. Opetus- ja tutkimuslaboratorio Technobothnia.

1.2.1 VAMK

Vaasan ammattikorkeakoulu on monikulttuurinen osakeyhtiömuotoinen korkeakoulu, joka kouluttaa tradenomeja, sosionomeja, sairaanhoitajia, terveydenhoitajia ja insinöörejä kansainvälisiin liike-elämän ja teollisuuden työtehtäviin. Paikallisten yritysten tarpeet vaikuttavat koulutuksen tarjontaan ja toteutukseen hyvähenkisen yhteistyön kautta.

Vaasan ammattikorkeakoulussa on kolme koulutusala, joihin kuuluu sosiaali- ja terveysala, liiketalous ja tekniikka. Opiskelijoita koulussa on noin 3300 ja henkilöstöä noin 210, joista 124 on opettajia.

Koulutusta Vaasan ammattikorkeakoulussa saa sekä suomen että englannin kielillä ja opetuksen kesto on 3,5-4 vuotta, riippuen valitusta tutkinnosta. Tutkinnot koostuvat perus- ja ammattiopinnoista, suuntaavista opinnoista, vapaasti valittavista opinnoista, opinnäytetyöstä sekä työharjoittelusta. Työharjoittelu ja opinnäytetyö pyritään tekemään koulutusta vastaavassa yrityksessä, mutta tarpeen vaatiessa myös ammattikorkeakoulu saattaa tarjota mahdollisuuden niiden suorittamiseen. /2/

1.2.2 Technobothnia

Opetus- ja tutkimuslaboratorio Technobothnia (Kuva 1) perustettiin vastaamaan lisääntyntä yhteistyötä Vaasan insinöörikoulutuksen ja seudun yritysten välillä. Laboratoriot sijaitsevat vuonna 1996 kunnostetuissa Vaasan Puuvilla Oy:n käytössä olleessa noin 8000 neliön kokoisessa tehdaskiinteistössä. Technobothnialla on monipuolisine laitekantoinen varustettuja laboratorioita ja opetustiloja sekä insinööri- että diplomi-insinöörikoulutukseen.

Technobothnia on Vaasan ammattikorkeakoulun, Vaasan yliopiston ja Yrkeshögskolan Novian yhteinen tilahanke, jossa insinööriopiskelijat suorittavat laboratorioharjoituksia teknillisissä opetusaiheissa. Tutkimus- ja kehittämishankkeiden lisäksi Technobothnia on tarjonnut testi-, mittaus- ja koepalveluita sekä koulutusta alueen teollisuudelle ja elinkeinoelämälle jo yli 20 vuotta. /3/

2 TAUSTATIETOA OPINNÄYTETYÖN AIHEESTA

2.1 Teollisuusrobotti

Jokaisella henkilöllä on oma mielikuva siitä mikä robotti on. Useimmat henkilöt, jotka ovat nauttineet elokuvista, televisiosarjoista, tieteiskirjallisuudesta ja videopeleistä saattavat ajatella robotteja ihmiskuntaa kapinoivina koneina. Tekniikan parissa työskentelevä henkilö taas ajattelee väsymätöntä työkonetta tehtaalla tai varastossa. Näitä jälkimmäisiä koneita kutsutaan teollisuusroboteiksi.

Teollisuusrobotit ovat sähköisesti, paineilmalla tai hydraulisesti liikkuvia tietokoneohjattuja laitteita, joiden liikeradat ovat muokattavissa ohjelman avulla. Yleinen robotin määritelmä on vähintään kolme ohjelmoitavaa liikeakselia ja yksi työkalu, mutta kyseinen määritelmä ei ole tänä päivänä enää tarkka. Automatisoidut ajoneuvot, kuten vihivaunut AGV (Autonomously Guided Vehicle) ja AIV (Autonomous Indoor Vehicle) ovat myös luokiteltavissa roboteiksi.

Yleiset teollisuusrobotit voidaan jakaa mekaniikkansa suhteen nivelvarsirobotteihin, lineaarisesti liikkuviin portaalirobotteihin, rinnakkaisrakenteisiin robotteihin ja SCARA -robotteihin (Kuva 2). /4/



Kuva 2. Yleisimmät teollisuusrobotit. Robotit vasemmalta oikealle: nivelvarsirobotti, rinnakkaisrakenteinen robotti, lineaarinen portaalirobotti ja SCARA -robotti.

2.1.1 Robottityypit

Nivelvarsiroboteissa on 5-6 liikeakselia, joiden ansiosta ne ovat joustavia ja monikäyttöisiä. Näiden ominaisuuksien ansiosta nivelvarsirobotit ovat nousseet teollisuudessa suosituiksi usealla sovellusalueilla, kuten kokoonpano-, pakkaus-, hitsaus- ja maalaustehtävissä. /4/

Portaaliroboteissa on kolme vapausastetta (DOF) yleensä suorakulmaisessa koordinaatistossa. Yksinkertaisen rakenteensa ansiosta portaalirobotit kestävät erinomaisesti suuriakin kuormituksia. Yleiset käyttötarkoitukset portaalirobotille on robotisoidut radat, nosturitelineet ja 3D-tulostimet. /4/

Rinnakkaisrakenteinen robotti on työstötehtävissä ja poimintatehtävissä käytetty teollisuusrobotti, jossa työkalu on kolmen lineaarisen vapausasteen varassa muodostaen suljetun rakenteen. Ripustettava robotti on rakenteensa ansiosta kestävä ja erittäin nopea, mutta sillä on myös rajoittunut työalue. /5/

SCARA-robotissa on kaksi tai kolme samassa tasossa kiertyvää niveltä ja neljäs pystysuuntainen lineaarinen liike. SCARA-robotteja käytetään pääasiassa elektronikkateollisuuden kokoonpanotehtävissä sen tarkkuuden, pienen koon ja nopeiden liikkeiden ansiosta. /5/

2.1.2 Robottien työkalut

Työkalut ovat sähköisiä tai pneumaattisia toimilaitteita, joita robotti siirtää asemasta toiseen. Tavallisin työkalu on tarrain, mutta hitsauspistooli, maaliruisku ja sähkötyökalut ovat myös yleisiä. Tarraimet ovat mekaanisia, magneettisia ja alipainetta hyödyntäviä työkaluja, joiden tehtävänä on kuljettaa kappaleita paikasta toiseen. /5/

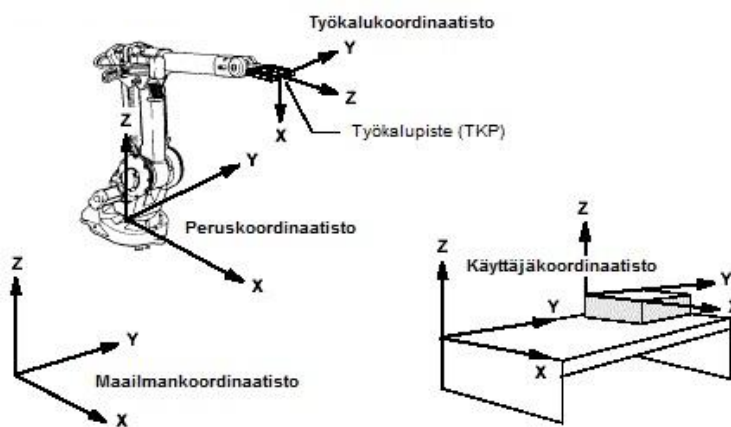
2.1.3 Ohjausjärjestelmä

Robotin ohjaimet ovat suuritehoisia prosessitietokoneita, jotka pystyvät ohjaamaan jokaista robotin niveltä tuhansia kertoja sekunnissa. Tämä on erityisen tärkeää kun halutaan saavuttaa suoraviivainen liike ohjaamalla niveliä oikeaan suun-

taan oikealla nopeudella toisiinsa nähden. Robotin ohjain koostuu keskusyksiköstä, ohjelmamuistista, käsiohjaimesta ja akselikohtaisista paikoitusjärjestelmistä. Sen tehtävä on tulkita ohjelma liikekäskyiksi ja ohjata liikeakseleita. Tämän lisäksi ohjain tarkkailee ympäristöä antureiden avulla, kuten törmäyksiä ja robotin toimintakuntoa. /4/

2.1.4 Koordinaatistot

Koordinaatisto on geometrinen järjestelmä, jolla kuvataan pisteiden paikoitusta avaruudessa koordinaattien avulla. Kolmiulotteisessa avaruudessa olevalla pisteellä on kolme koordinaattia, joita kuvataan usein nimikkeillä X, Y ja Z. Koordinaatiston keskustassa on origo, jonka kaikki koordinaatin arvot ovat nolla. Teollisuusrobottien yhteydessä käytettyihin koordinaatistoihin lukeutuu peruskoordinaatisto, maailmankoordinaatisto, työkalukoordinaatisto ja käyttäjäkoordinaatisto (Kuva 3).



Kuva 3. Robotiikassa käytetyt koordinaatistot.

Peruskoordinaatisto (Base) on robotin jalustaan kiinnitetty koordinaatisto jonka X- ja Y-akselit muodostavat vaakasuoran tason robotin jalustan alapinnan tasolle. Z-akseli on pystyakseli, joka kulkee ensimmäisen liikeakselin pyörähdyskeskipisteen kautta.

Maailmankoordinaatisto (World) on robotin toimintaympäristöön kiinnitetty koordinaatisto, joka määräytyy ympäristössä olevasta laitteistosta tai rakenteista. Ohjelmoinnin kannalta maailmankoordinaatiston käyttö on kannattavaa silloin, kun robotin liikkeet määräytyvät ulkoisten laitteiden perusteella.

Työkalukoordinaatisto (Tool) on koordinaatisto, jonka origo on yleensä sidottu työkalun koordinaatistiksi. Tämän ansiosta koordinaatiston akseleiden orientaatio mukautuu työkalun mukana. /4/

2.1.5 Ohjelmointi

Online -ohjelmointi sisältää johdattamalla ohjelmoinnin ja opettamalla ohjelmoinnin. Johdattamalla ohjelmoiminen tapahtuu vapauttamalla robotin liikeakselit, jonka jälkeen ohjelmoiva henkilö liikuttaa robottia lihasvoimin haluttua liikerataa pitkin. Robotin ohjain muistaa suoritettua liikeradan ja pystyy toistamaan liikkeen. Johdattamisessa on kuitenkin huonona puolena ohjelman hankala muuttaminen, jonka takia tapa on yleisempi varsinkin yhteistyörobottien kanssa, jotka ovat rakenteellisesti kevyempiä. Ohjelmoiminen opettamalla on huomattavasti yleisempi ohjelmointimenetelmä, jossa robottia liikutetaan käsiohjaimella haluttuun paikkaan ja paikka tallennetaan ohjaimen muistiin. Tallennetuille pisteille määritetään liikekomento, joka määrittää liiketavan ja nopeuden pisteeseen. Pisteisiin voidaan lisätä myös I/O-käskyjä, joilla hallitaan esimerkiksi toimilaitteita. /4/

Offline -ohjelmointi, eli etäohjelmointi tarkoittaa robotin ohjelmointia ulkoisen ohjelmiston avulla. Näissä ohjelmistoissa on 3D-mallit roboteista, jotka toimivat kuten oikeat robotit. Offline ohjelmointi on nykyään yleinen ohjelmointitapa, koska robottia ei tarvitse pysäyttää uuden liikeradan opettamisen ajaksi. Kuitenkin jos etäohjelmoinnista halutaan saada kaikki irti, kannattaa ohjelmoinnissa käyttää CAD -jäljitelmää solusta. /4/

ABB -robottien ohjelmoinnissa käytetään RAPID -ohjelmointikieltä (Kuva 4) käskyjen luomiseen. RAPID -ohjelmointikielen ohjelma koostuu päärutiinista, alirutiineista ja ohjelmadatasta. Käsky määrittää toiminnon, kuten robotin liikku-

misen tai hypyn ohjelman sisällä, ja ne suoritetaan siinä järjestyksessä jossa ne on ohjelmoitu. Yleisimmät käskyt, joita käytetään robotin liikuttamiseen ovat: MoveL, MoveJ ja MoveC. MoveL on lineaariliike, MoveJ on nivel-liike ja MoveC on kaariliike. /4/

```
MODULE MainModule
  CONST robtarget Koti:=[[226.26113075,-
320.803097442,211.901154713],[-0.000000001,-0.000000058,1,0.000000004],
[-1,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget p_1:=[[226.261108398,-320.803131104,5],[0,0,1,0],
[-1,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget p_2:=[[107.493164063,-313.168334961,5.000061035],
[0,0,1,0],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget p_4:=[[228.873535156,-399.716796875,5.000366211],
[0,0,1,0],[-1,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
  PROC main()
    kolmio;
  ENDPROC
PROC kolmio()
  MoveJ Koti,v400,z0,t_pencil\WObj:=wobj0;
  MoveL p_1,v400,z0,t_pencil\WObj:=wobj0;
  MoveL p_2,v400,z0,t_pencil\WObj:=wobj0;
  MoveL p_4,v400,z0,t_pencil\WObj:=wobj0;
  MoveL p_1,v400,z0,t_pencil\WObj:=wobj0;
  MoveL Koti,v400,z0,t_pencil\WObj:=wobj0;
ENDPROC

ENDMODULE
```

Kuva 4. Esimerkki Rapid -ohjelmointikielestä. /6/

2.2 3D-CAD

Tietokoneavusteisella suunnittelulla (CAD) tarkoitetaan suunnitelman luomista, muokkaamista, analysointia tai optimointia tietokoneohjelman avulla. Suunnitelmalla voidaan tarkoittaa teknisiä piirustuksia ja 3D-malleja. 3D-mallinnus otettiin monissa yrityksissä käyttöön kaksiulotteisen suunnittelun tilalle, koska kolmiulotteisten kappaleiden muokattavuus on huomattavasti helpompaa. Malli antaa myös visuaalisen ylliotteen aikaisemmasta menetelmästä. /7/

2.2.1 3D-ohjelmistot

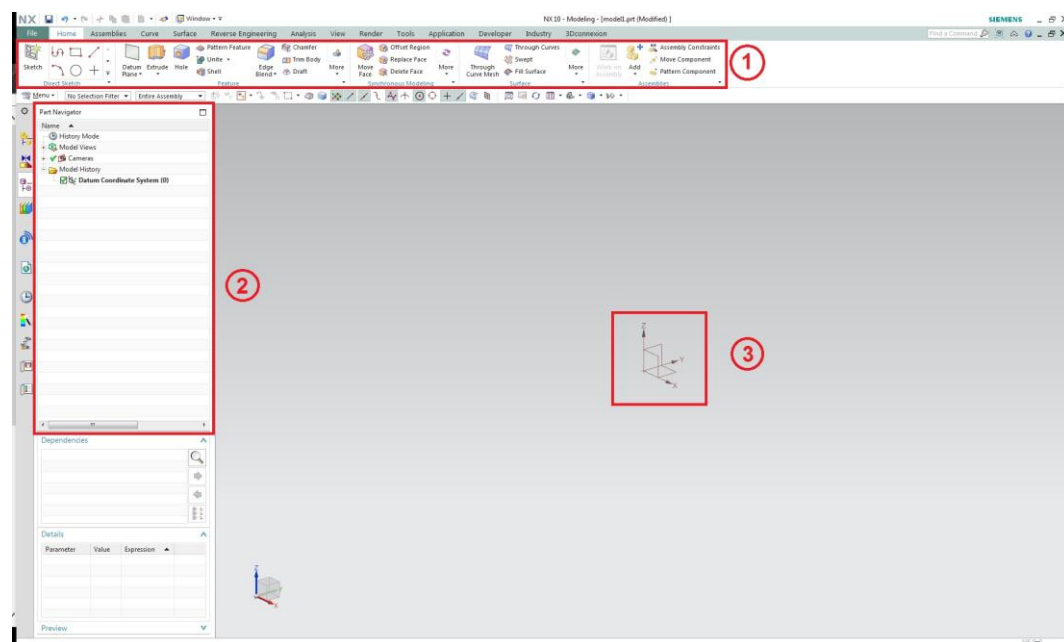
3D-suunnitteluun suunnattuja ohjelmistoja on saatavilla melkoisen suuri määrä, joissa jokaisessa on omat hyvät ja huonot puolet. Saatavana olevat ohjelmistot,

kuten kaikki muukin, jakautuvat halpoihin ja kalliisiin versioihin. Edullisten ja hintavien ohjelmistojen suurin ero näkyy niiden ominaisuuksissa, sillä hintavissa ohjelmistoissa on enemmän integroituja mahdollisuuksia valmistukseen ja analysointiin mallinnuksen lisäksi. /8/

Suosituimmat mallintajat ovat:

- SolidWorks
- Autodesk Inventor
- Siemens NX
- CATIA
- VERTEX

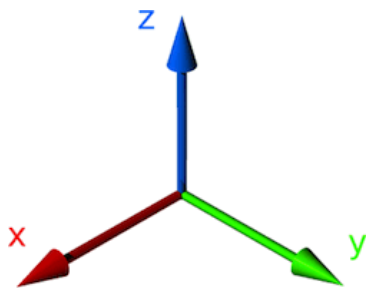
Vaikka mallintajia löytyykin suuri määrä, niin kaikissa mallintajissa on suurin piirtein samat perustoiminnot. Tämän takia sopivan mallintajan löytäminen riippuu yrityksen tarpeista ja työntekijöiden kokemuksesta. /8/



Kuva 5. Siemens NX 10.0 -mallinnusohjelmisto, johon kuuluu toimintovalikko (1), mallipuu (2) ja avaruuden origo (3).

2.2.2 Mallintaminen

Mallintaminen tapahtuu kolmiulotteisessa avaruudessa, joka koostuu X-, Y- ja Z-koordinaattiakseleista. Nämä koordinaattiakselit muodostavat origon (Kuva 6), joka toimii mallinnuksen lähtöpisteenä. /9/



Kuva 6. Koordinaatistojen keskipiste, eli origo.

Mallintaminen on käytännössä erilaisten mitoitettujen piirteiden yhdistämistä yksi kerrallaan halutun kokonaisuuden saavuttamiseksi. Ennen piirteiden määrittämistä, on kannattavaa (mutta ei välttämätöntä) laatia luonnos, joka toimii mallinnuksen pohjana. Kun luonnos on valmis ja pursotettu, se toimii mallin aihiona. Aihioon voi lisätä tai poistaa materiaalia piirteiden ja operaatioiden avulla. Yleisimmät piirteet, joita käytetään mallinnuksessa ovat: pursotus, reikä, ryhmäkopiointi, pyöristys ja kuori. Näiden piirteiden käyttö toimivan ja tehokkaan lopputuloksen saavuttamiseksi vaatii luovaa ajattelua, suunnittelutaitoa sekä harjoittelua. Mitat, joita käytetään piirteiden määrittämiseen, kutsutaan parametreiksi. /8/

Parametrien hallitseminen on luultavasti yksi tärkeimmistä taidoista, jota tarvitaan mallintamisen aikana. Parametrit ovat määritettyjä mittoja tai lausekkeita, jotka vaikuttavat piirteiden muotoon ja kokoon. Piirteiden mittoja voidaan muokata helposti *expressions* -toiminnolla, jota käyttämällä myös riippuvuussuhteiden ylläpitäminen on helpompaa.

Kappalekohtaisia riippuvuussuhteita voidaan luoda lausekkeilla, jolloin piirteen arvo vaikuttaa toisen piirteen arvoon. Riippuvuussuhteita rakentamalla koko mallin rakennetta on mahdollista muokata muuttamalla vain muutamaa arvoa. Nämä

arvot ovat yleensä mallin fyysisiä mittoja, kuten halkaisija, pituus, leveys tai/ja korkeus. Toinen riippuvuussuhde on piirteiden hierarkia, jossa koordinaatisto toimii vanhempana uudelle luonnokselle, joka puolestaan toimii vanhempana piirteelle. Tätä riippuvuussuhdetta kutsutaan äiti/lapsi-suhteeksi. Suhteita voidaan käyttää myös kokoonpanoissa, joissa arvon muuttaminen voi vaikuttaa muihin malleihin. /8/

2.2.3 3D-mallit

Mallinnuksia käytetään laajasti tuotesuunnittelussa, tuotevalmistuksessa, markkinoinnissa ja videopeleissä. 3D-mallinnusohjelmistot käyttävät omia tiedostomuotoja mallien tallentamiseen, mikä tekee mallintajien väliset vaihdokset vaikeiksi. Tämän takia vaihdoksissa käytetään usein standardisoitua STEP -formaattia. STEP -tiedostomuoto mahdollistaa mallin toimivuuden koko tuotteen elinkaaren ajan ohjelman vaihdoksista huolimatta.

2.3 Simulaatio ja virtuaalitodellisuus

Simulointi on todellisuuden jäljittelyä, jossa toimenpiteen kulkua ja lopputuloksia tarkkaillaan turvallisesti keinotekoisia malleja hyödyntäen. Luodut mallit esittävät simulointikohteen käytöstä, ominaisuuksia ja toimintoja. Simulaatiolla voidaan tarkoittaa fysikaalista simulointia (kuten mekanisoitu sonni) tai tietokonesimulointia, jossa tarkkailu tapahtuu virtuaalisessa ympäristössä. Myös mielikuvitus on eräänlaista simulointia. Simulointia käytetään tekniikan parissa tuottavuuden optimointiin, turvallisuuden kehittämiseen, testaukseen ja koulutukseen.

Tietokonesimulaatio on kasvanut suosituksi menetelmäksi, sillä se on edullinen tapa tarkkailla tilanteita, jotka ovat vaarallisia tai joita ei yksinkertaisesti vielä ole. Mutta kuten kaikella muullakin simuloinnilla on myös heikkouksia, kuten saadun tiedon luotettavuus. Simulaatiossa käytetään paljon oletuksia ja arvioita, jotka tekevät tuloksista epätarkkoja. Tietokonesimulaatioissa käytetään realistisia fysiikkamoottoreita, kun halutaan saada mahdollisimman tarkkoja tuloksia. /10/

Virtuaalitodellisuus on nopeasti kehittyvä tekniikka, jossa tietokoneen, kahden sensorin, kahden ohjaimen ja virtuaalitodellisuuslasien kanssa (Kuva 7) pyritään

luomaan todellisuutta jäljittelevä tunnelma. Virtuaalitodellisuudessa käyttäjä pysyy katsomaan ympäristöä ensimmäisessä persoonassa, kuin hän olisi oikeasti paikalla. Ohjaimien avulla käyttäjä voi liikkua, käyttää valikkoa ja vaikuttaa ympäristöönsä vaikka liikuttamalla esineitä. /11/



Kuva 7. HTC Vive -virtuaalitodellisuuspaketti.

2.3.1 Simulaatio ja virtuaalitodellisuus robotiikassa

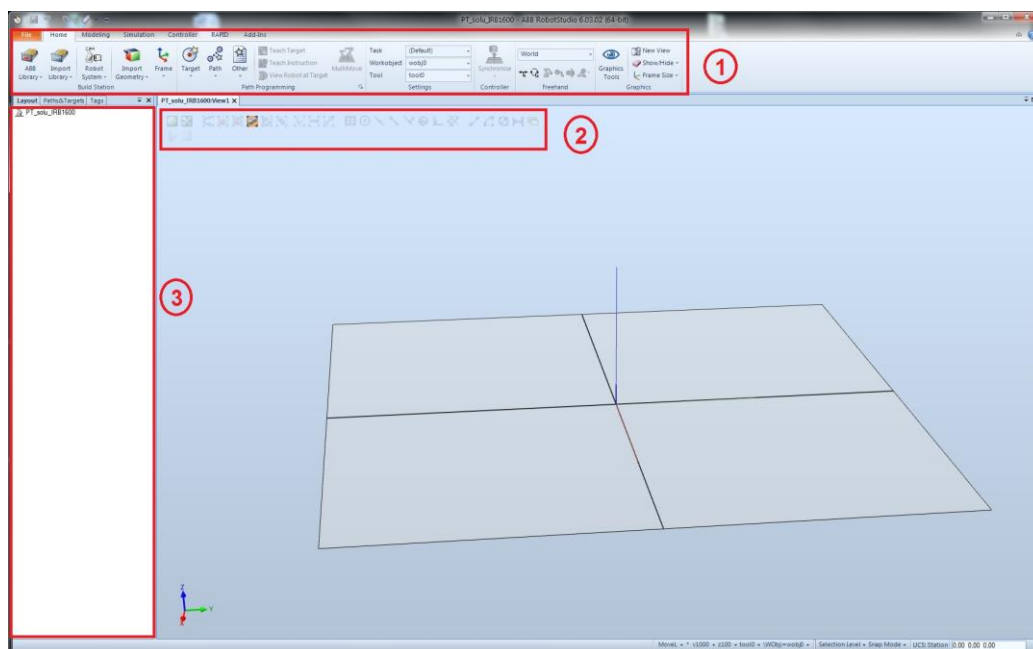
Robottisimulaattori on tietokoneohjelma, jonka avulla voidaan etäohjelmoida ja simuloida robotin liikkeitä. Ohjelmassa on VirtualControlleri, jota myös oikea robotti käyttää, joten simuloitu robotti käyttäytyy samalla tavalla kuin oikea robotti. Robotin simuloinnilla saavutetaan taloudellinen tapa selvittää tarvittavat liikera-dat, robotin tuottavuus ja solun valmistuskustannukset ennen robotin hankkimista. /12/

Virtuaalitodellisuus on nopeasti saavuttanut automaation ja robotiikan alat, sillä se tarjoaa ennennäkemättömiä mahdollisuuksia etäohjelmoinnissa. Virtuaalitodellisuus yhdistää käytännössä kaikki kolme ohjelmointimenetelmää, jotka mainittiin aikaisemmin. ABB oli ensimmäinen yritys, joka tarjosi mahdollisuuden ohjelmoida robotteja virtuaalitodellisuuden avulla. Uutuutensa ansiosta virtuaalitodelli-

suustekniikka myös lisää ohjelmoijan mielenkiintoa ohjelmoinnin tarkastamiseen, säätämiseen ja optimisointiin. /13/

2.3.2 Simulointiympäristö

RobotStudio (Kuva 8) tarjoaa simulaatiota varten ympäristön, johon virtuaalinen solu rakennetaan. Ohjelmisto sisältää kaikki tarpeelliset toiminnot etäohjelmointiin kaikissa ohjelmoinnin eri vaiheissa, kuten lähestymisliikkeiden opetuksessa ja olemassa olevan ohjelman optimoinnissa.



Kuva 8. RobotStudion simulointiympäristö.

Toimintovalikko ja sen välilehdet (1) sisältävät kaikki ohjelmiston toiminnot, joita käytetään solun ohjelmoimiseen. *Home* -välilehdestä löytyvät toiminnot, joita käytetään solun rakentamiseen ja liikeratojen luomiseen. *Modeling* -välilehdessä olevilla toiminnoilla voidaan luoda ja ryhmittää komponentteja. *Simulation* -välilehden toiminnoilla hallitaan ohjelmiston simulointia koskevia ominaisuuksia.

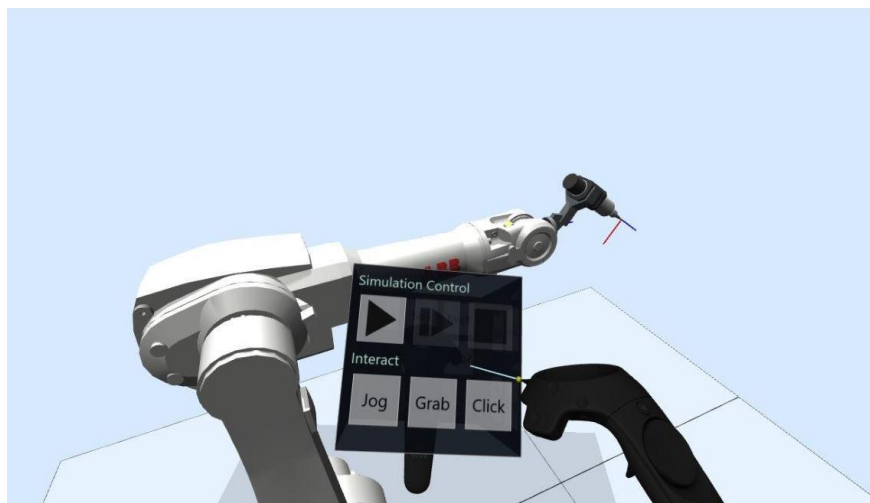
Controller -välilehti sisältää toiminnot VirtualControllerin hallitsemiseen. *Rapid* -välilehti sisältää työkalut ohjelman muokkaamiseen.

Kuvakevalikossa (2) ohjelmoija voi valita kuvakulman, määrittää valinta-asetukset ja käyttää erilaisia mittausvaihtoehtoja. Valikosta löytyy myös simuloinnin aloitus- ja lopetuspainikkeet.

Sivuvalikkoon (3) ilmestyvät solun sisältö, liikeradat ja erilaiset valintaikkunat. Solun sisältöluettelolla voidaan yksinkertaisesti muokata solun komponenttien ominaisuuksia, kuten nimiä, värejä ja paikoituksia. Myös robotin asteittain liikuttaminen onnistuu valikosta löytyvän toiminnon avulla.

2.3.3 Virtuaaliympäristö

Virtuaaliympäristö (Kuva 9) on avoin avaruus, jossa voi liikkua vapaasti. Ympäristöstä löytyy RobotStudiassa rakennettu solu, jota voidaan manipuloida virtuaalisesti. Manipulointi tapahtuu ohjaimilla, joista oikeanpuoleisella valitaan toiminnot ja liikutetaan komponentteja kun taas vasemman puoleisessa ohjaimessa on valikko. Valikko sisältää kaikki virtuaalitodellisuudessa käytössä olevat toiminnot, kuten simuloinnin hallinta, välitön liikkuminen, tarttuminen ja liikeradan opettaminen.



Kuva 9. RobotStudion virtuaaliympäristö.

3 SOLUN ETÄOHJELMOINTIYMPÄRISTÖ

3.1 Robottisolun mallintaminen

Etäohjelmointiympäristön mallintaminen aloitettiin robottisolun mitoittamisella, jonka jälkeen työ eteni mallintamiseen. Solun mallintamisen suunnittelu lähti yksinkertaisesta tavoitteesta, eli mahdollisimman realistisen jäljennöksen luomisesta. Kaikki mallien muodot ja yksityiskohdat eivät olleet tarpeellisia käyttötarkoitusta ajatellen, mutta ne lisättiin siitä huolimatta.

3.1.1 Mitoittaminen

Mitoittamisen tärkeys jää usein layout-suunnittelussa vähäiseksi, koska lopullinen layout rakennetaan suunnitelman pohjalta. Projektin aikana valmistettava etäohjelmointiympäristö perustuu olemassa olevaan soluun, joten mitoittamisella on huomattavasti suurempi rooli ja se tulee suorittaa huolella, sillä mitoitusvirheet saattavat johtaa kaluston vaurioitumiseen.

Mitoituksessa käytettiin pääasiassa mittanauhaa, mutta tarpeen vaatiessa, eteenkin pienien komponenttien kanssa käytettiin työntömittaa. Luonnollisesti mittanauhaa käyttämällä mittauksen tulokset vaihtelivat ja toleranssit jäivät ajoittain suuriksi. Tarkemman lopputuloksen saavuttamiseksi solun komponenttien kriittiset mitat jouduttiin tarkistamaan useita kertoja. Kriittisiin mittoihin kuuluu robotin läheisyydessä olevien komponenttien pituus, leveys ja varsinkin korkeus.

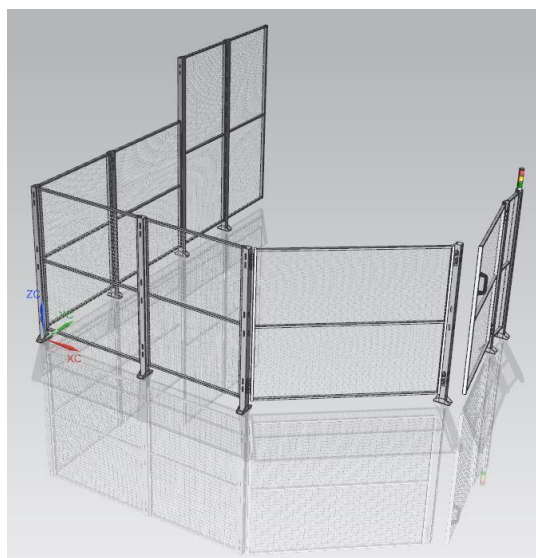
3.1.2 3D-CAD -mallit

Solun mitoittamisen jälkeen siirryttiin komponenttien mallintamiseen. Mallinnuksessa käytettiin luonnollisesti mittaustuloksia apuna jotta lopputuloksesta tulisi mahdollisimman todenmukainen. Mallintamisessa käytettiin Siemens NX 10.0 -mallinnusohjelmistoa, jonka käyttö oli tullut tutuksi opintojen aikana. NX tarjoaa laajan arsenaalin ominaisuuksia, jotka tekevät mallintamisesta sujuvaa ja ajoittain haasteellista sopivin määrin.



Kuva 10. Siemens NX 10.0:n toimintovalikko.

Aitauksen (Kuva 11) mallintaminen aloitettiin pylväistä, jotka pitävät aidat pystyssä. Pylväitä oli kaksi eri kokoa: 1400 mm ja 2000 mm, jotka kuvaavat pylväiden korkeuksia. Pylväiden muut mitat olivat kaikissa samat. Aitauksen runko mallinnettiin seuraavaksi ja ritilät lisättiin ryhmäkopiointi-toimintoa apuna käyttäen, koska toiminto on hyödyllinen tapa kopioida geometriaa ja pitää mallinnuspuu siistimpänä. Parametrien käyttäminen oli tärkeää aitojen kanssa, sillä aitoja oli useampi koko. Parametreja käyttämällä mallinnusta voitiin muuttaa sen sijaan, että jokainen aita mallinnettiin erikseen. Realistisuuden nimissä aitaukseen mallinnettiin myös kolme eri saranaa ja turvavalo. Suurin osa layoutin komponenttien ominaisuuksista, kuten valot, kytkimet ja saranoiden liikkeet eivät valitettavasti toimi etäohjelmointiympäristössä, vaan ne on mallinnettu pelkästään ulkonäön vuoksi. RobotStudiossa on varmasti tapa saada toiminnot käyttöön, mutta työn tarkoitusta ajatellen ne eivät ole välttämättömiä.



Kuva 11. Mallinnetun solun aitaus.

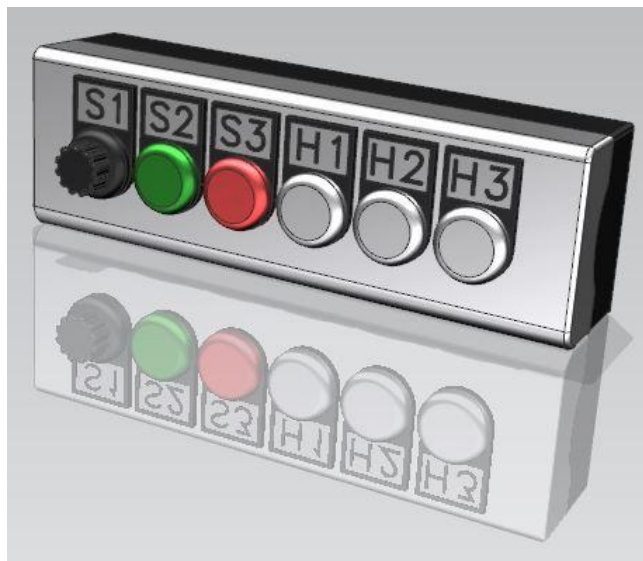
Jalustojen ja pöytien mallintaminen oli työvaiheen helpoimpia, sillä niiden muodot ovat yksinkertaiset. Pöytien mallintaminen aloitettiin aina pöydän tasosta, koska taso on oltava oikean kokoinen ja oikealla korkeudella. Kun taso on mallinnettu oikealle korkeudelle, jalat lisättiin malliin. Jalustojen mallintaminen suoritettiin samaa käytäntöä noudattaen.

Kuljettimet olivat huomattavasti haasteellisempia mallintaa, koska niissä oli paljon eri osia, jotka täytyi mallintaa yksitellen. Työvaihetta helpotti hieman kun alumiiniprofiilien mallintamisessa käytettiin parametreja apuna, joten kaikkia profiileja ei tarvinnut mallintaa erikseen. Alumiiniprofiilien jälkeen mallinnettiin lisätarvikkeet, kuten jalat, ohjauskytkimet, konenäkö ja tietysti itse hihna. Hihnan mallintaminen vaati erityistä tarkkuutta, jotta hihna sopi rullien ympärille.



Kuva 12. Yksi mallinnetun solun kuljettimista.

Toimilaitteen mallintaminen oli ehdottomasti haasteellisin vaihe. Itse työkalun muoto ei ollut haastava, mutta mallintamisessa vaadittu tarkkuus teki työstä vaativan. Työkalu päätettiin mallintaa kokoonpanona, jotta toimilaitteen yksittäiset osat olisivat mahdollisimman tarkat. Tämä ratkaisu vaati soveltamista, sillä osien mitoissa oli heittoa. Lopulta osat saatiin yhdistettyä hyväksyttäväksi kokonaisuudeksi, mutta tulevaisuudessa mallinnustekniikkaa saa pohtia tarkemmin.



Kuva 13. Mallinnetun solun turvakytin.

Loput mallinnetuista komponenteista olivat erilaisia ja kaikenmuotoisia komponentteja, jotka mallinnettiin yksinomaan yksityiskohtien takia. Näitä komponentteja olivat esimerkiksi kytkimet, tuolit ja jakkarat, seinät ja torni. Näiden komponenttien mallintamisessa käytettiin samoja tekniikoita kuin aikaisemmin mainituissa komponenteissa, joten niiden mallintamista ei käsitellä enempää.

3.1.3 Ratkaisut mallien keventämiseksi

Yleinen ongelma mallintamisessa on piirteiden suuri määrä mallinnuspuussa, koska niiden määrä aiheuttaa helposti ongelmia mallinnusohjelmistossa ja hidastaa tietokonetta. Monimutkaisissa ja suurissa muodoissa piirteiden määrä on luonnollisesti suurempi, joten määrän pienentäminen vaatii usein mallintajalta luovaa ajattelua.

Eräs hyödyllinen keino mallien keventämiseksi on kokoonpanojen käyttäminen. Kokoonpanojen käyttäminen kokoonpanoissa pienentää piirteiden määrää mallin-
nusuissa ja siten helpottaa tietokoneen työtä. Keinon haittapuoli on tiedostojen
kasvava määrä. Jokainen kokoonpano on uusi tiedosto ja tämä helposti hukuttaa
kohdekansion ja haluttua tiedostoa voi olla vaikeaa löytää.

3.2 CAD -mallien siirto etäohjelmointiympäristöön

Ennen mallien siirtämistä etäohjelmointiympäristöön, mallit täytyy muuttaa tuet-
tuun tiedostomuotoon. RobotStudion tuettu tiedostomuoto on ACIS, joka tunne-
taan myös SAT -tiedostona. Valitettavasti NX ei tallenna tiedostoja kyseiseen
muotoon, joten mallinnukset täytyi muuttaa toisen mallinnusohjelman kautta. En-
nen mallinnusohjelman vaihtamista, mallit täytyi muuttaa STEP -tiedostoiksi, jot-
ta ne voidaan avata toisessa mallintajassa.

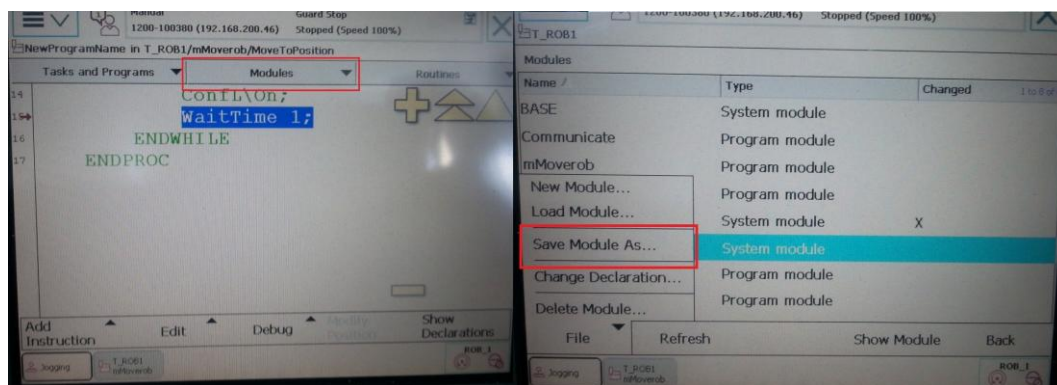
3.2.1 Komponenttien viimeistely

Komponenttien viimeistely onnistui helposti RobotStudiassa. Viimeistelyyn kuu-
lui värien muuttaminen vastaamaan todellisia värejä (SAT -tiedostomuoto ei tue
esimääritettyjä värejä), nollapisteen asettaminen ja työkalun luominen. Kompo-
nentti lisätään ympäristöön painamalla *Import Geometry* -painiketta ja valitsema-
lla *Browse for Geometry*.



Kuva 14. Työkalupisteen kalibroitiprosessi.

Työkalun luominen aloitettiin tarkastamalla työkalupisteiden tarkkuus. Tarkastaminen tapahtuu orientoimalla robotin työkalua ja seuraamalla, että työkalun kärki pysyy paikallaan (Kuva 14). Jos työkalun kärjen paikka vaihtelee orientaation aikana, TKP pitää määrittää uudelleen. Määrittäminen tapahtuu paikoittamalla työkalun kärki kiinteään pisteeseen useasta lähestymispisteestä. Pisteiden lukumäärän voi valita itse, mutta on huomioitavaa, että useampi piste johtaa tarkempaan tulokseen. Työtä suoritettaessa TKP:n tarkkuudet olivat hyväksyttävät, joten kalibrointia ei tarvinnut suorittaa. Kun työkalujen tarkkuus oli tarkastettu, oli aika ladata TKP:n tiedot robotilta.



Kuva 15. Työkalutietojen lataaminen robotilta.

Työkalutiedot löytyvät *Jogging* -välilehdestä, jossa valitaan *Modules*. Uudesta ikkunasta valitaan *File > Save Module As...* > valitaan tiedostolle kohde ja nimi. Kohteeksi voidaan valita USB -tikku, joka voidaan kiinnittää *FlexPendanttiin*.

```

MODULE user (SYSMODULE)

! Predefined user data
! *****

! Declaration of numeric registers reg1...reg5
VAR num reg1 := 0;
VAR num reg2 := 0;
VAR num reg3 := 0;
VAR num reg4 := 0;
VAR num reg5 := 0;

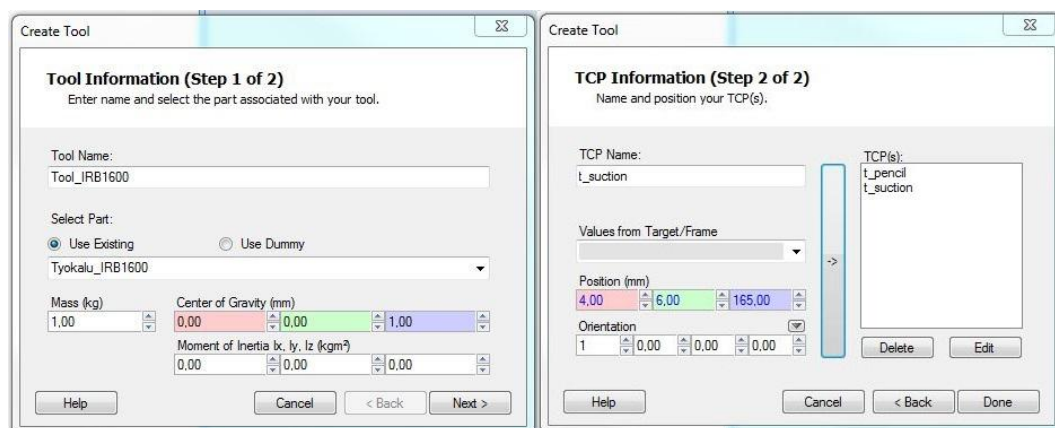
! Declaration of stopwatch clock1
VAR clock clock1;

PERS tooldata t_pencil:= [TRUE, [[87.1946, -60.8708, 88.3598],
[0.684703, 0.139054, 0.702623, -0.134784]], [0.5, [0, 0, 50],
[1, 0, 0, 0], 0, 0, 0]];
PERS tooldata t_pencil2:= [TRUE, [[70.7156, 96.4691, 85.8945],
[0.598166, -0.370411, 0.591581, 0.393732]], [0.5, [0, 0, 50],
[1, 0, 0, 0], 0, 0, 0]];
PERS tooldata toolstat:= [FALSE, [[714.514, -60.6976, 1042.4],
[1, 0, 0, 0]], [1, [0, 0, 20], [1, 0, 0, 0], 0, 0, 0]];
PERS wobjdata pahvi:= [TRUE, TRUE, "", [[2.62911, 97.4427, 34],
[0.0193709, 0.973327, -0.228562, -0.00437374]], [[0, 0, 0],
[1, 0, 0, 0]]];
TASK PERS tooldata suction:= [TRUE, [[4, 6, 165], [1, 0, 0, 0]], [1,
[0, 0, 50], [1, 0, 0, 0], 0, 0, 0]];
TASK PERS tooldata gripper:= [TRUE, [[0, 0, 250], [1, 0, 0, 0]], [1,
[0, 0, 100], [1, 0, 0, 0], 0, 0, 0]];
TASK PERS tooldata gripper_new:= [TRUE,
[[78.1637, 119.54, 129.596], [0.689807, -0.691017, 0.152611, -
0.152878]], [1, [0, 0, 100], [1, 0, 0, 0], 0, 0, 0]];

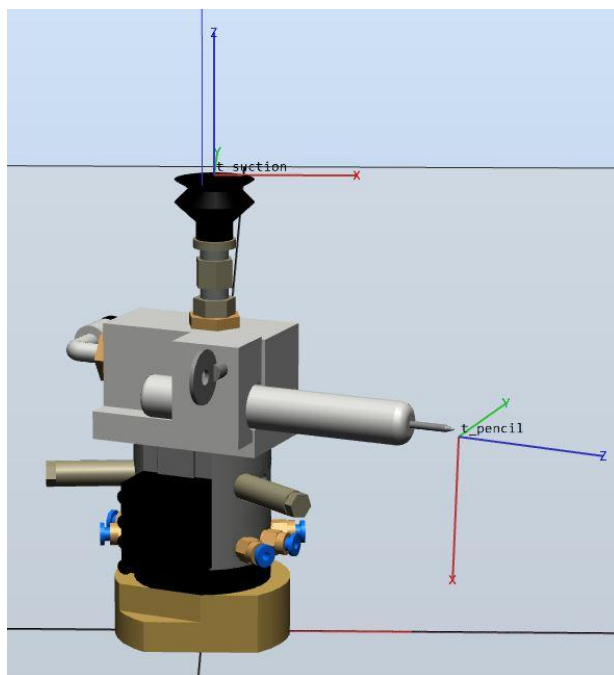
```

Kuva 16. Ote työkalutiedoista. Toimilaitteen työkalujen tiedot korostettuna punaisella.

Kun TKP:n tiedot oli hankittu robotilta, niin tiedot syötettiin RobotStudiassa olevaan työkaluun. Tietoihin kuului TKP paikoitus koordinaatteina suhteessa robotin laippaan ja orientaatio kvaternioina. Aluksi valittiin *Create Tool* -toiminto, jonka jälkeen työkalulle voitiin määrittää geometria, eli tiedosto, nimi, massa ja painopiste.



Kuva 17. Työkalutietojen syöttäminen toimilaitteen mallille. Vasemmalla määritetään toimilaitteen nimi, tiedosto, massa ja painopiste. Oikealla annetaan työkalupisteiden nimet, paikoitus ja orientaatio.



Kuva 18. Työkalupisteiden koordinaatit toimilaitteen mallissa.

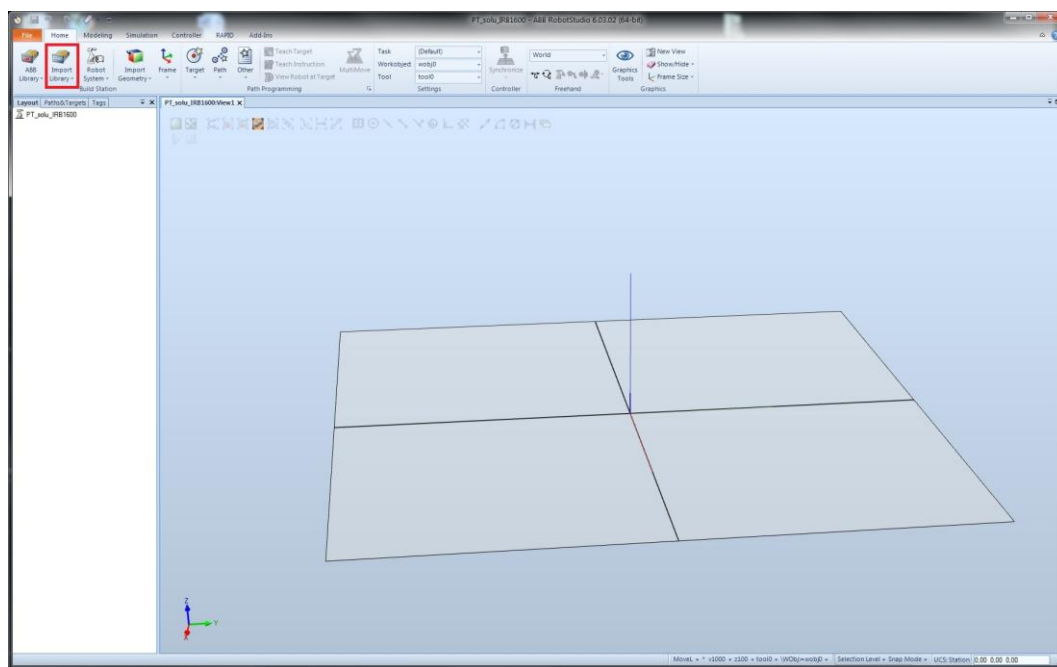
Suurin osa työkalun ja kuljettimien alustamisesta jäi vähäiseksi, koska toimilaitteet ja kuljettimet täytyy usein muuttaa SmartComponenteiksi simulointia varten. SmartComponentit ovat älykkäitä komponentteja, jotka voivat liikuttaa kappaleita

ja kommunikoida robotin kanssa. Simuloinnissa SmartComponentteja käytetään kuvastamaan antureita ja sensoreita, koska niitä voidaan ohjelmoida RobotStudioissa. SmartComponentin luominen on työkohtaista, joten se jätettiin tekemättä.

Kun komponentit oli alustettu, ne muutettiin RSLIB -tiedostoiksi. RSLIB -tiedosto on RobotStudio oma tiedostomuoto, joka säilyttää komponentille määritetyt ominaisuudet vaikka komponentin kohdesolun vaihtaisi. Tämä mahdollistaa komponentin lisäämisen uudessa solussa samaan paikkaan ympäristön koordinaatistossa.

3.2.2 Layout

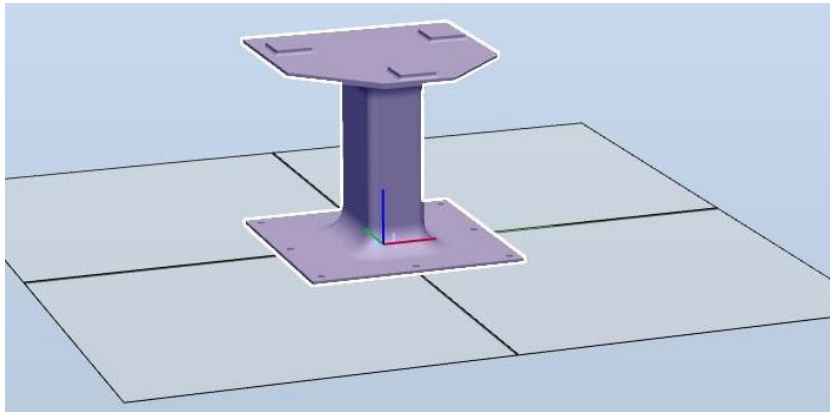
Layoutin rakentaminen aloitettiin, kun kaikki solun komponentit oli viimeistelty. Rakentaminen aloitettiin avaamalla RobotStudio ja valitsemalla *Create New Solution* nimeämisen ja kohdekansion valitsemisen jälkeen.



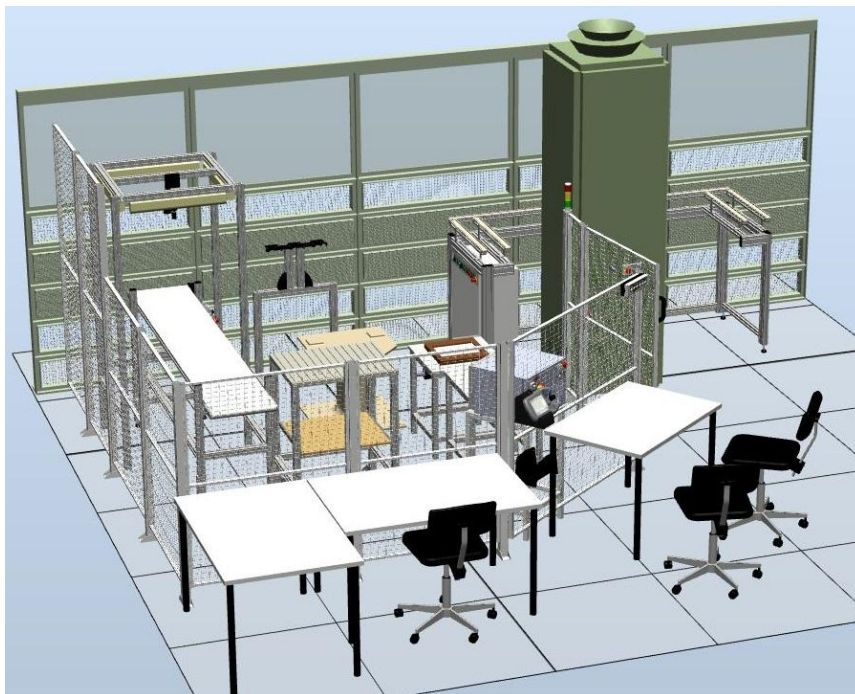
Kuva 19. Import library.

Kun uusi layout on auennut onnistuneesti, voidaan lisätä ensimmäinen komponentti. Komponentti lisätään painamalla *Import Library* ja valitsemalla *Browse for*

Library (Kuva 19). Jalusta valittiin ensimmäiseksi komponentiksi, koska se sijaitsee solun keskellä robotin alla. Jalusta paikoittuu automaattisesti keskelle solua, koska jalustan koordinaatisto sijaitsee keskellä sen alapinnassa. Jokainen komponentti ilmestyy koordinaatistoon origot sisäkkäin, ellei toisin ole määritetty alustusvaiheessa. Loput komponentit lisättiin mitoituksien mukaan (Kuva 21).

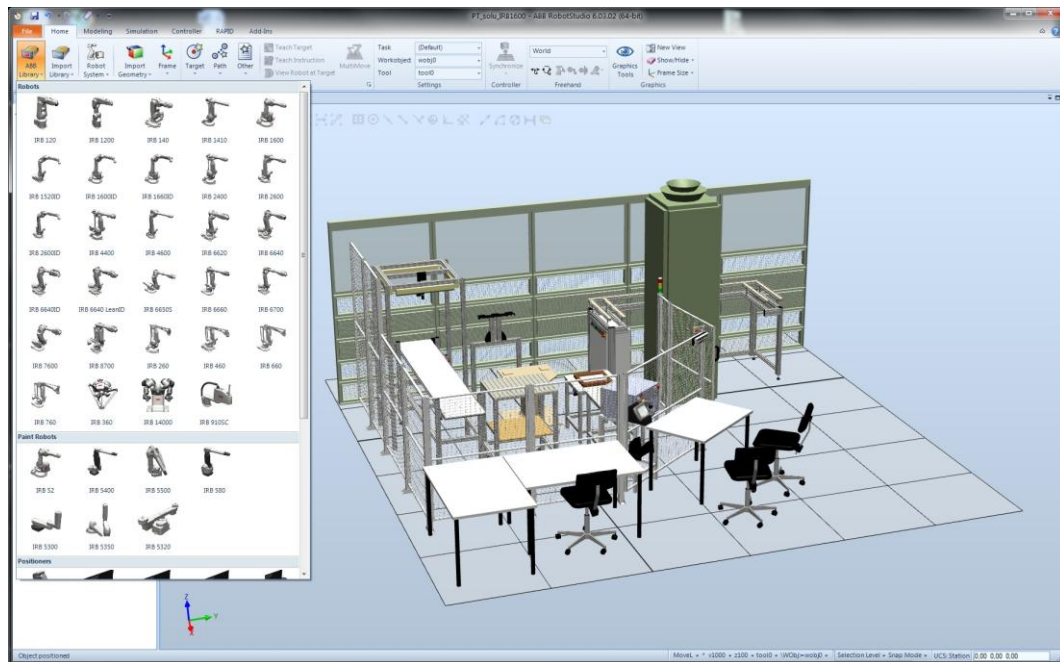


Kuva 20. Soluun lisätty robotin jalusta.

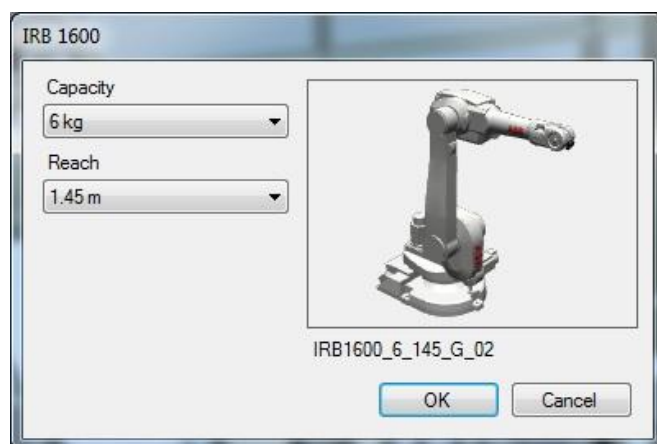


Kuva 21. Loput komponentit lisättynä soluun.

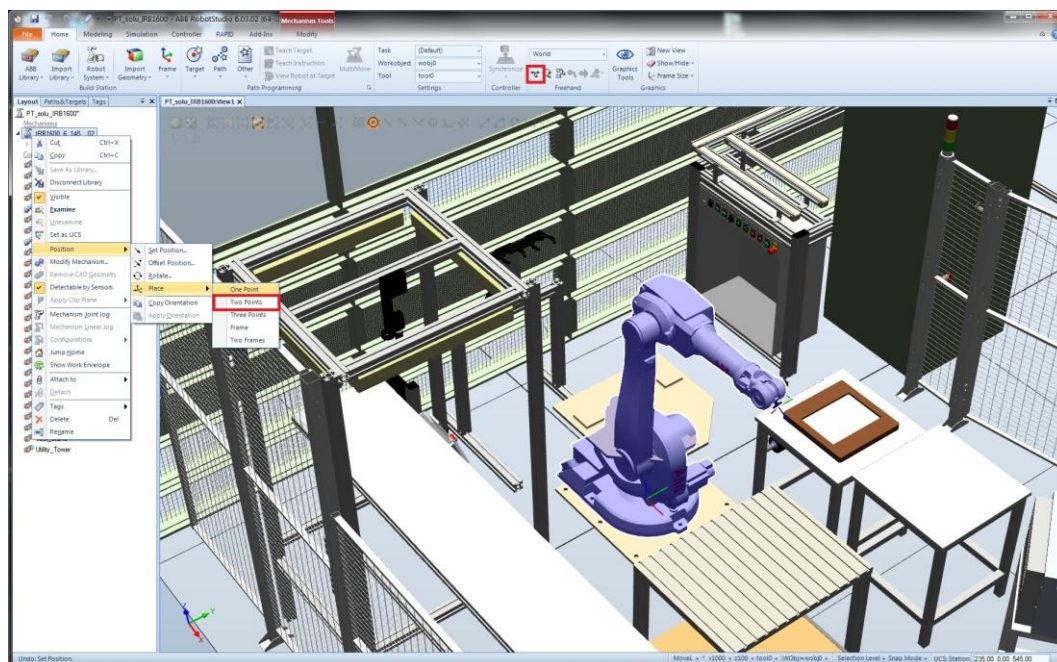
Kun solun komponentit ovat paikoillaan, lisätään robotti painamalla *ABB Library* ja valitsemalla haluttu robotti, joka työn solussa oli IRB 1600 (Kuva 22). Robotin valitsemisen jälkeen robotin voi määrittellä tarkemmin valitsemalla nostokapasiteetin ja ulottuvuuden (Kuva 23).



Kuva 22. ABB Library.

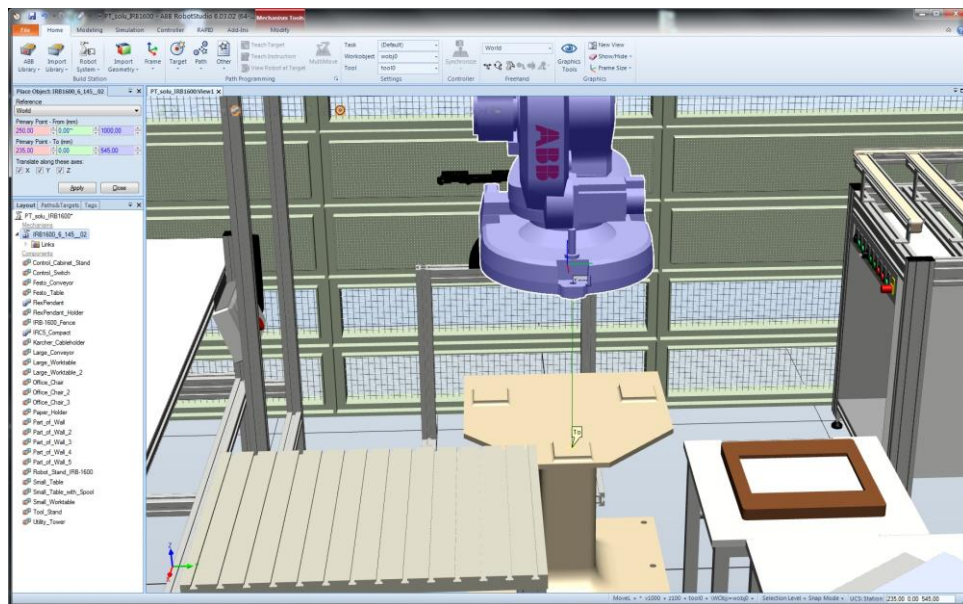


Kuva 23. Robotin kapasiteetin ja ulottuvuuden valinta.



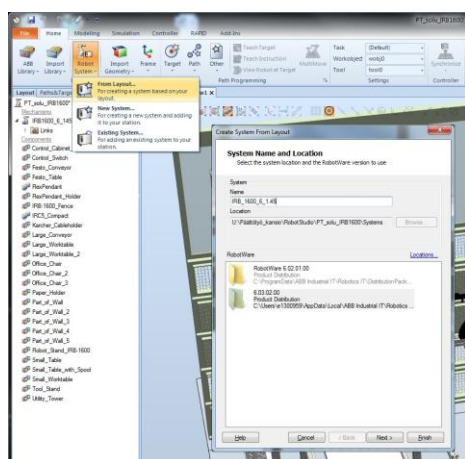
Kuva 24. Lisätyn robotin paikoittaminen 1/2. Komponentin lineaarisen liikkeen toiminto ja komponentin paikoituksen toiminto korostettuna.

Robotti ilmestyy maailmankoordinaatiston keskelle. Jotta robotin saa asetettua jalustan päälle paikalleen, robotti on nostettava jalustan yläpuolelle. Nosto tapahtuu valitsemalla robotin ja *linear move* -komennon, jonka jälkeen robotin voi vain vetää uuteen paikkaan. Tämän jälkeen asetetaan robotti paikalleen painamalla robotin ikonia layout-puusta ja kohdistamalla hiiri position -valikon päälle, jonka jälkeen kohdistetaan *Place* -valikon päälle ja valitaan *Two Points*. Uudesta valikosta valitaan alkupiste, joka on robotin kiinnitysreikä. Sitten valitaan kohdepiste, joka on jalustassa olevan vahvikkeen keskipiste. Valitsemalla *Apply*, robotti siirtyy haluttuun paikkaan.

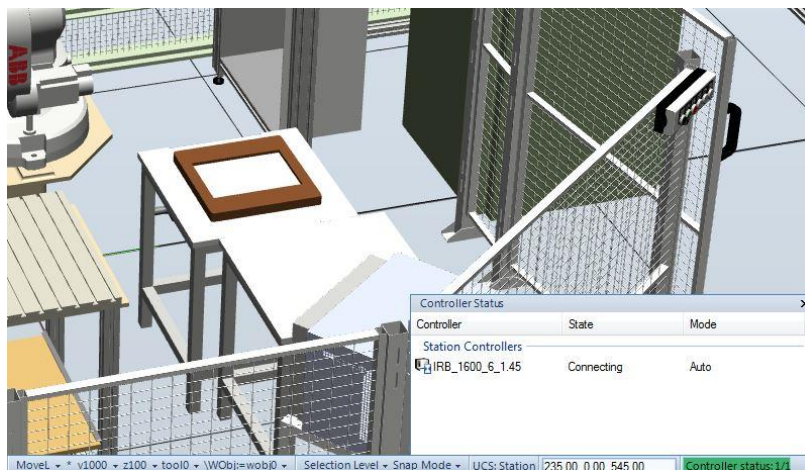


Kuva 25. Lisätyn robotin paikoittaminen 2/2.

Kun robotti on halutussa paikassa, VirtualControlleri voidaan lisätä layoutiin. Liisääminen tapahtuu painamalla Robot System -painiketta ja valitsemalla From Layout. Tämän jälkeen systeemille voidaan määrittää nimi ja kohdekansio. Kun VirtualControlleri on asetettu onnistuneesti RobotStudion oikeaan alakulmaan ilmestyy ikkuna, joka kertoo Controllerin olevan toiminnassa vihreällä värillä (Kuva 27).

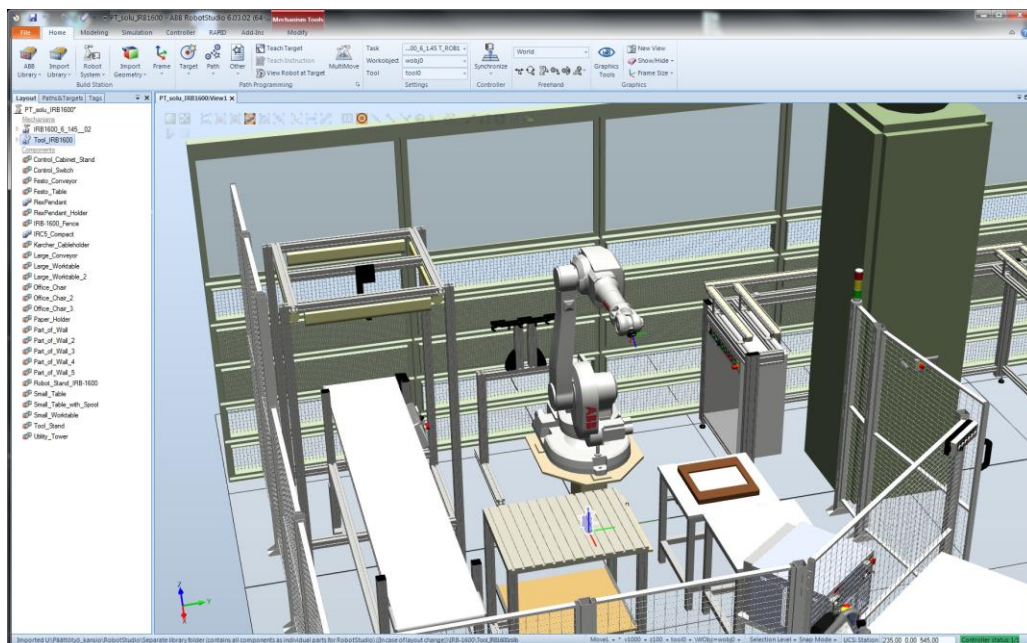


Kuva 26. VirtualControllerin luominen.

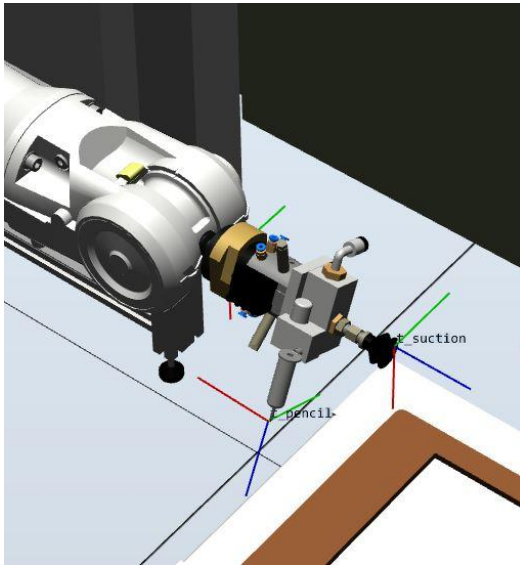


Kuva 27. Controllerin toiminnan ilmoittava ikkuna.

Lopuksi lisätään viimeistelty työkalu, joka ilmestyy koordinaatiston keskelle, kuten kaikki muutkin komponentit (Kuva 28). Työkalun saa yhdistettyä robottiin yksinkertaisesti vetämällä toimilaitteen ikoni robotin ikonin päälle. Kun ohjelma kysyy, että haluatko päivittää työkalun aseman niin valitaan kyllä. Työkalu on nyt kiinnitetty robottiin.



Kuva 28. Maailmankoordinaatiston keskelle ilmestynyt työkalu.



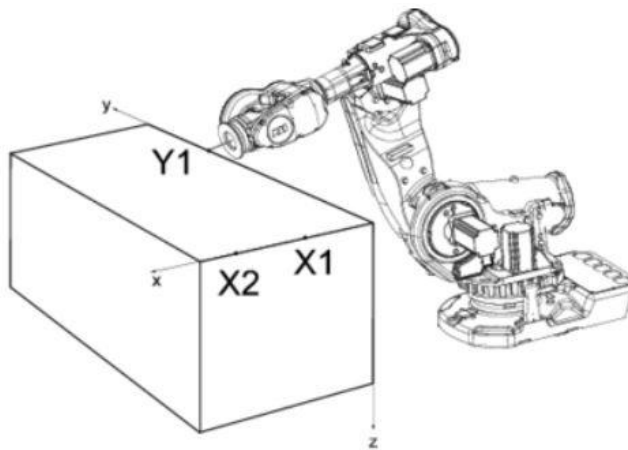
Kuva 29. Työkalu kiinnitettynä robottiin.

Etäohjelmointiympäristö on nyt valmis kalibroitavaksi.

3.3 Ympäristön kalibrointi

Ennen kun etäohjelmointiympäristöä voidaan käyttää, se kannattaa kalibroida. Kalibroimisella tarkoitetaan prosessia, jossa virtuaalinen, mallinnettu robottisolu asetetaan yhtenäiseksi alkuperäisen solun kanssa. Kalibrointi on tarpeellinen prosessi, jos etäohjelmointiympäristöltä vaaditaan tarkkuutta.

Helppo kalibroiminen tapahtuu luomalla käyttäjäkoordinaatisto, sillä RobotStudiossa ei ole varsinaista kalibrointityökalua. Käyttäjäkoordinaatisto on todellisuudessa vain referenssikoordinaatisto, jota käytetään robotin paikoitusten määrittämiseen, kun työalue muuttuu. Yleensä tämä koordinaatisto asetetaan kiinteään pisteeseen, kuten kappaleen tai pöydän kulmaan.



Kuva 30. Käyttäjäkoordinaatisto.

3.4 Etäohjelmointiympäristön siirto

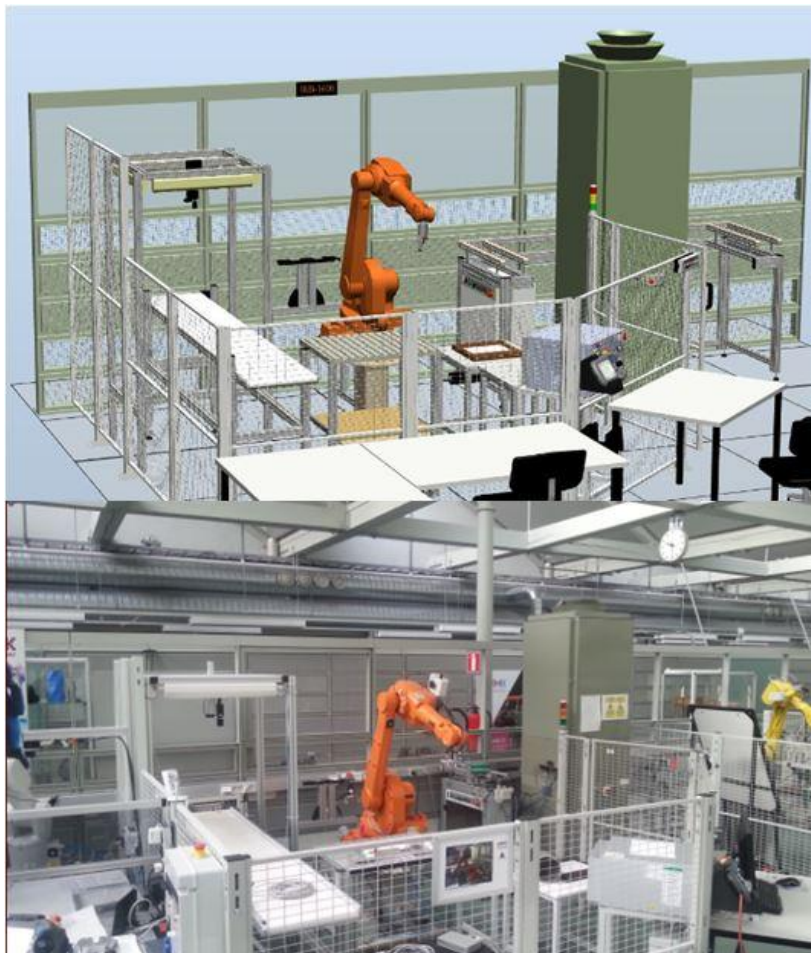
Jotta etäohjelmointiympäristö voidaan siirtää onnistuneesti toiselle koneelle, se täytyy pakata *pack&go* -toiminnolla, joka löytyy *share* -välilehdestä. Pakatessa solua zip -tiedostoon, lisätään kaikki tiedostot, joista solu koostuu. Kun solu pakataan RobotStudiassa, ohjelmisto tarjoaa mahdollisuuden määrittää salasana zip -tiedostolle. Salasanan käyttö ei ole pakollista, mutta se tarjoaa hieman turvaa yrityksille, jotka haluavat pitää työnsä salassa.

Kun ympäristö on pakattu onnistuneesti, se voidaan siirtää toiselle koneelle valitulla tavalla, kuten sähköpostilla, pilvipalvelulla tai muistitikulla. Kun ympäristö on siirretty toiselle koneelle, ympäristö voidaan avata *unpack&work* -toiminnolla. Toiminto purkaa kansion ja tallentaa sisällön haluttuun kohteeseen, jonka jälkeen työntekoa voidaan jatkaa normaalisti.

4 POHDINTAA

4.1 Työkatsaus

Työssä luotiin tarkka jäljennös olemassa olevasta robottisolusta käyttämällä Siemens NX 10.0 -mallinnusohjelmistoa ja RobotStudio -ohjelmistoa. Työ vaati perehtymistä erilaisiin mallinnustekniikoihin, sekä RobotStudion käyttöön simuloinnissa ja virtuaalitodellisuudessa. Suurin osa opinnäytetyöstä kului solun komponenttien mitoittamisessa ja mallintamisessa. Etäohjelmointiympäristö on tarkoitus tulla käyttöön opetuksen yhteydessä Vaasan ammattikorkeakoulussa.



Kuva 31. Rakennettu virtuaalinen solu ja oikea solu vierekkäin.

Etäohjelmointiympäristö tarjoaa monia hyötyjä, mutta kuten kaikessa muussakin, myös tuotetussa etäohjelmistoympäristössä on haittansa. Yksi näistä haitoista on solun tarkkuudesta johtuva geometrioiden suuri määrä, joka saattaa olla työtä vaikeuttavaa tai hämmentävää ohjelmoijalle. Ympäristön käyttäminen vaatii kuvakulmien jatkuvaa vaihtamista, sillä mallinnetut komponentit voivat olla ohjelmoijan tiellä ja täten vaikeuttaa työtä. Valmis etäohjelmointiympäristö saavuttaakin täyden potentiaalinsa vasta, kun ympäristöä käytetään virtuaalitodellisuudessa.

Vaikka soluun tulikin muutoksia lähes välittömästi työn valmistumisen jälkeen, olen melko tyytyväinen työn lopputulokseen. Työ tarjosi sopivasti haasteita ja tuntui sopivan pituiselta.

4.2 Jatkokehitys

Vaikka solu saatiin toimintavalmiuteen ja se vaikuttaa näyttävältä, niin työtä voidaan aina kehittää pidemmälle ja työstää paremmaksi. Työn suorittamisen jälkeen, ja oikeastaan sen suorittamisen aikanakin, tuli mieleen pari kohdetta, joita voisi kehittää pitemmälle.

Eräs vartenotettava kehitysmahdollisuus on interaktiivisuuden lisääminen. Kuten aikaisemmin mainittiin, solun aitaus, napit ja kytkimet eivät toimi etäohjelmointiympäristössä. Vaikka näiden komponenttien toiminta ei ole solun toiminnan kannalta välttämätöntä, niin avattava aitauksen ovi lisäisi solun realistisuutta, varsinkin virtuaalitodellisuudessa. Jos interaktiivisuutta päätetään lähteä kehittämään, niin virtuaalitodellisuuden käyttäminen vaatisi suuremman tilan, jotta solussa voi kävellä normaalisti. Tällä hetkellä laajempaa tilaa tai kehittynyttä interaktiivisuutta ei tarvita, sillä virtuaaliympäristössä liikutaan välittömästi teleportoimalla.

Toinen kehityksen kohde on virtuaalitodellisuuden ja lisätyn todellisuuden yhdistäminen. Ajatuksena olisi, että ympäristö tarjoaa tietoa robotin toiminnasta, solun tapahtumista tai mahdollisia virheraportteja esimerkiksi näkökenttään tulevassa ikkunassa. Valitettavasti vain RobotStudio tai vastaava ohjelmisto kykenee toteuttamaan tämän mahdollisuuden.

LÄHTEET

- /1/ Hakala, P., Mannila, M. & Verkkonen, V. 2016. Opinnäytetyön mallipohja. VAMK opetusmateriaali.
www.puv.fi/fi/study/opintoihin_osallistuminen/opinnaytetyo/opinnayte_mallipohja.rtf
- /2/ VAMK. 2017. Vaasan ammattikorkeakoulun kotisivu. Viitattu 19.6.2017.
<http://www.puv.fi/fi/about/>
- /3/ Technobothnia. 2017. Technobothnian kotisivu. Viitattu 19.6.2017.
<http://www.technobothnia.fi/>
- /4/ Keinänen, T. , Kärkkäinen, P. , Lähetkangas, M. & Sumujärvi, M. 2010. Automaatiojärjestelmien logiikat ja ohjaustekniikat. Helsinki: WSOY Oppimateriaalit Oy.
- /5/ Robotiikka. Lahden Ammattikorkeakoulu. 2016. Rajoitettu opetusmateriaali. Viitattu 21.11.2017.
http://miniweb.lpt.fi/automaatio/opetus/luennot/pdf_tiedostot/Robotiikka_yleinen.pdf
- /6/ Billing, M. 2012. IRC-5 Yleisohje. Vaasa: VAMK opetusmateriaali.
- /7/ Narayan, K., Rao, K. & Sarcar, M. 2008. Computer Aided Design and Manufacturing. New Delhi: Prentice Hall of India.
- /8/ Hantula, J. 2015. 3D-Mallinnus. Vaasa: VAMK opetusmateriaali.
- /9/ Tuhola, E. & Viitanen, K. 2008. 3D-mallintaminen suunnittelun apuvälineenä. Jyväskylä: Tammertekniikka.
- /10/ Wikipedia, Simulation. Viitattu 22.11.2017.
<https://en.wikipedia.org/wiki/Simulation>
- /11/ Wikipedia, Virtual Reality. Viitattu 22.11.2017.
https://en.wikipedia.org/wiki/Virtual_reality
- /12/ Wikipedia, Robotics Simulator. Viitattu 24.11.2017.
https://en.wikipedia.org/wiki/Robotics_simulator
- /13/ RobotWorx, ABB offers VR integration for robot programming. Viitattu 26.11.2017. <https://www.robots.com/blog/viewing/abb-offers-vr-integration-for-robot-programming>

LIITE 1. Komponenttien paikoitukset solun maailmankoordinaatistossa:

IRB 1600

	Position			Orientation		
	X	Y	Z	X	Y	Z
Robot_Stand_IRB-1600	0,0	0,0	0,0	0,0	0,0	90,0
Tool_Stand	-405,0	150,0	0,0	0,0	0,0	90,0
Festo_Table	1220,0	-824,0	902,0	0,0	0,0	90,0
Small_Table	1162,0	630,0	0,0	0,0	0,0	0,0
Small_Table_with_Spool	640,5	550,0	0,0	0,0	0,0	-90,0
Paper_Holder	460,5	725,2	657,0	0,0	0,0	0,0
Festo_Conveyor	-948,0	1045,0	0,0	0,0	0,0	0,0
Large_Conveyor	1500,0	-1435,0	0,0	0,0	0,0	90,0
Control_Cabinet_Stand	1682,0	510,0	0,0	0,0	0,0	-53,0
IRC5_Compact	2313,0	435,6	769,4	0,0	0,0	127,0
IRB1600_5_145_01	-15,0	0,0	545,0	0,0	0,0	0,0
IRB-1600_Fence	2525,0	-1795,5	0,0	0,0	0,0	90,0
Control_Switch	1632,4	1389,5	1322,0	0,0	0,0	127,0
Utility_Tower	-184,0	1921,0	0,0	0,0	0,0	90,0
Karcher_Cableholder	2500,0	359,4	476,8	0,0	0,0	127,0
FlexPendant_Holder	2600,0	438,0	870,0	0,0	0,0	127,0
FlexPendant	2478,1	452,6	978,8	-35,0	0,0	127,0
Large_Worktable_1	3314,7	-1025,9	0,0	0,0	0,0	90,0
Large_Worktable_2	2514,1	-1830,3	0,0	0,0	0,0	0,0
Small_Worktable	2828,3	985,8	0,0	0,0	0,0	127,0
Office_Chair_1	3045,0	-651,5	0,0	0,0	0,0	0,0
Office_Chair_2	2955,8	1102,0	0,0	0,0	0,0	-12,0
Office_Chair_3	2700,0	1600,0	0,0	0,0	0,0	55,0
Part_of_Wall_1	-1130,0	-1985,0	0,0	0,0	0,0	90,0
Part_of_Wall_2	-1130,0	-790,0	0,0	0,0	0,0	90,0
Part_of_Wall_3	-1130,0	405,0	0,0	0,0	0,0	90,0
Part_of_Wall_4	-1130,0	1600,0	0,0	0,0	0,0	90,0
Part_of_Wall_5	-1130,0	2795,0	0,0	0,0	0,0	90,0